

Lista de Exercícios - Métodos de busca e comparação de algoritmos

BCC702 - Programação de Computadores II

2025.1

INSTRUÇÕES

- A atividade é individual.
 - Implemente **TODOS** os exercícios.
 - Desenvolver o código e realizar os testes em C++.
 - Incluir comentários no código explicando as etapas. Comente trechos do código, não linha a linha!
 - Envie suas soluções no Moodle. Para enviar, crie um arquivo compactado com todas as suas soluções e envie o arquivo compactado.
 - Para questões que criem algum arquivo, não é necessário enviar o arquivo gerado.
 - Exercícios enviados fora do prazo serão penalizados com 10% da nota por dia de atraso.
-

Exercício 1

Altere os algoritmos de ordenação *Selection Sort* e *Insertion Sort* para ordenar um conjunto de números inteiros em ordem decrescente. Use os algoritmos apresentados na aula.

Considere que o tamanho máximo do vetor digitado será 10. Portanto, não é necessário alocar o vetor dinamicamente.

Exemplo 1

Tamanho do vetor: 6
Elementos: 5 3 12 8 2 7

Vetor ordenado:
12 8 7 5 3 2

Exercício 2

Crie um registro `Pessoa`, que contém os campos `nome` e `idade`. Faça um programa que lê do usuário um vetor de `Pessoas` e então ordena esse vetor. **Aqui é necessário alocar o vetor dinamicamente.** Sua ordenação deve primeiro considerar a ordem alfabética para os nomes e depois, para pessoas com nomes iguais, sua idade (a pessoa mais velha deve aparecer primeiro).

Para comparar se duas strings são iguais, já vimos que, **em C++**, podemos usar o operador `==`. Além dele, também podemos, **em C++**, usar os operadores `<`, `>`, `<=` e `>=`, da mesma forma como usamos para comparar valores numéricos.

Existem outros métodos, como a função `compare`¹, que vocês podem usar, caso queiram. Ela é usada da seguinte forma:

```
string str1 = "Teste", str2 = "teste";  
int cmp = str1.compare(str2);
```

`compare` retorna 0 se as strings são iguais, um valor estritamente negativo se `str1` é menor que `str2` e um valor estritamente positivo se `str1` é maior que `str2`. Porém, para o intuito desta atividade, os operadores `<`, `>`, `<=` e `>=` são suficientes.

É recomendável implementar uma função que compara dois valores do tipo `Pessoa`. Utilize o seguinte protótipo:

```
int comparaPessoa(Pessoa a, Pessoa b);
```

A função deve retornar 0 se as duas pessoas forem iguais, 1 se *a* for maior que *b* e -1 se *a* for menor que *b*.

Exemplo 1

```
Total de pessoas: 4  
Informe os dados da pessoa: Ana 20  
Informe os dados da pessoa: Carlos 30  
Informe os dados da pessoa: Beatriz 25  
Informe os dados da pessoa: Ana 35
```

```
Vetor ordenado:  
Ana 35  
Ana 20  
Beatriz 25  
Carlos 30
```

¹Veja em <https://cplusplus.com/reference/string/string/compare/>

Exemplo 1

Total de pessoas: 5
Informe os dados da pessoa: maria 18
Informe os dados da pessoa: João 22
Informe os dados da pessoa: joao 20
Informe os dados da pessoa: maria 40
Informe os dados da pessoa: Pedro 19

Vetor ordenado:

João 22
Pedro 19
joao 20
maria 40
maria 18

Exercício 3

Apresente o funcionamento dos métodos de ordenação por seleção e por inserção para ordenar as chaves a seguir:

3	4	9	2	5	1	8
---	---	---	---	---	---	---

Desenhe o vetor várias vezes demonstrando os passos de cada algoritmo. Coloque um círculo nos elementos movimentados e um traço entre os elementos ordenados e desordenados.